

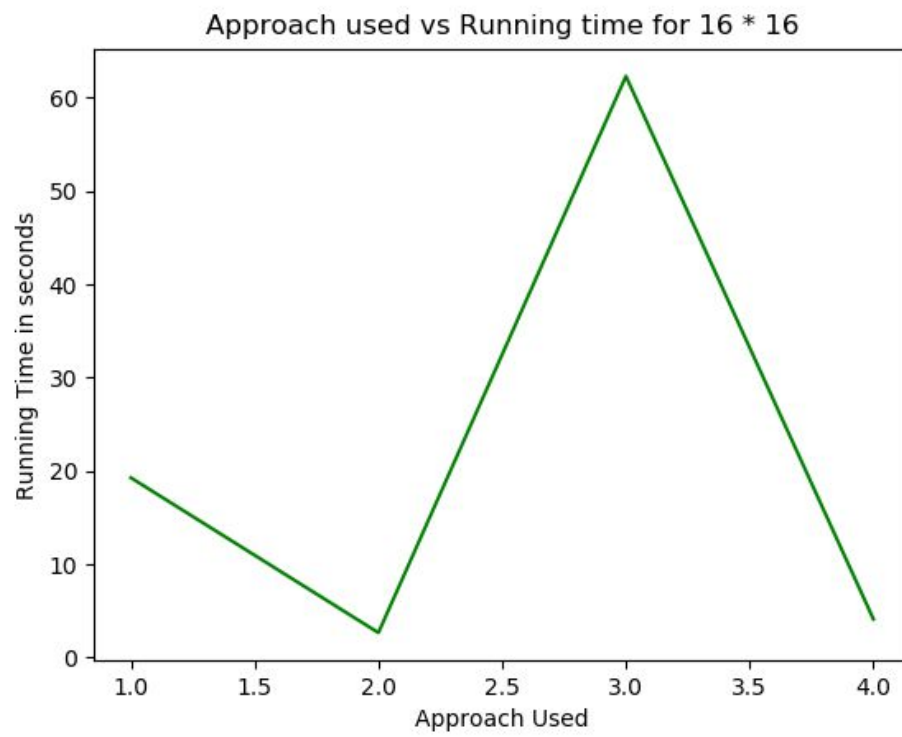
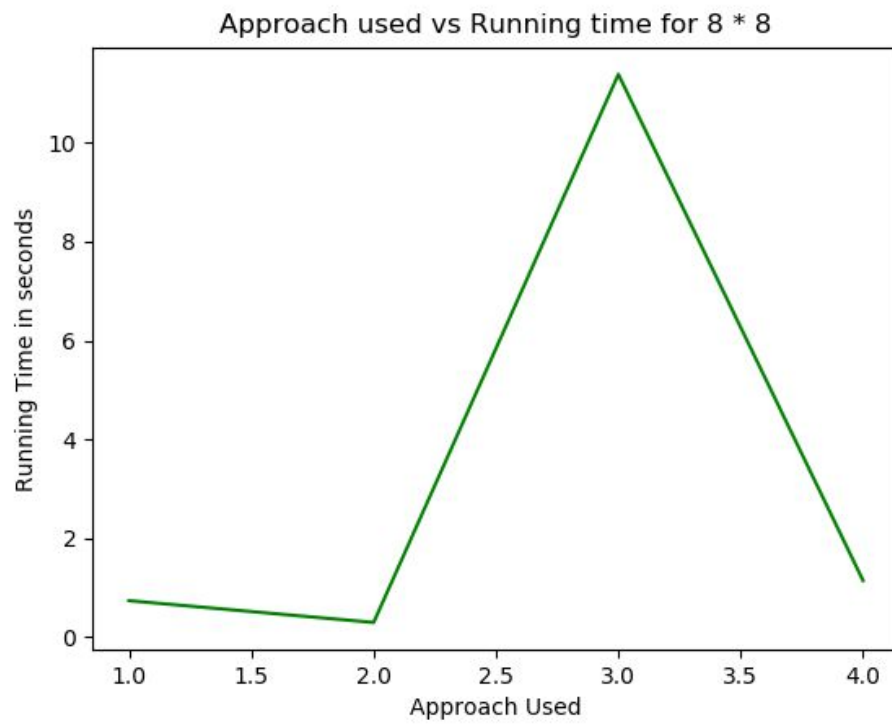
CS 535 Homework 2 by Aashish Adhikari and Parijat Bhatt

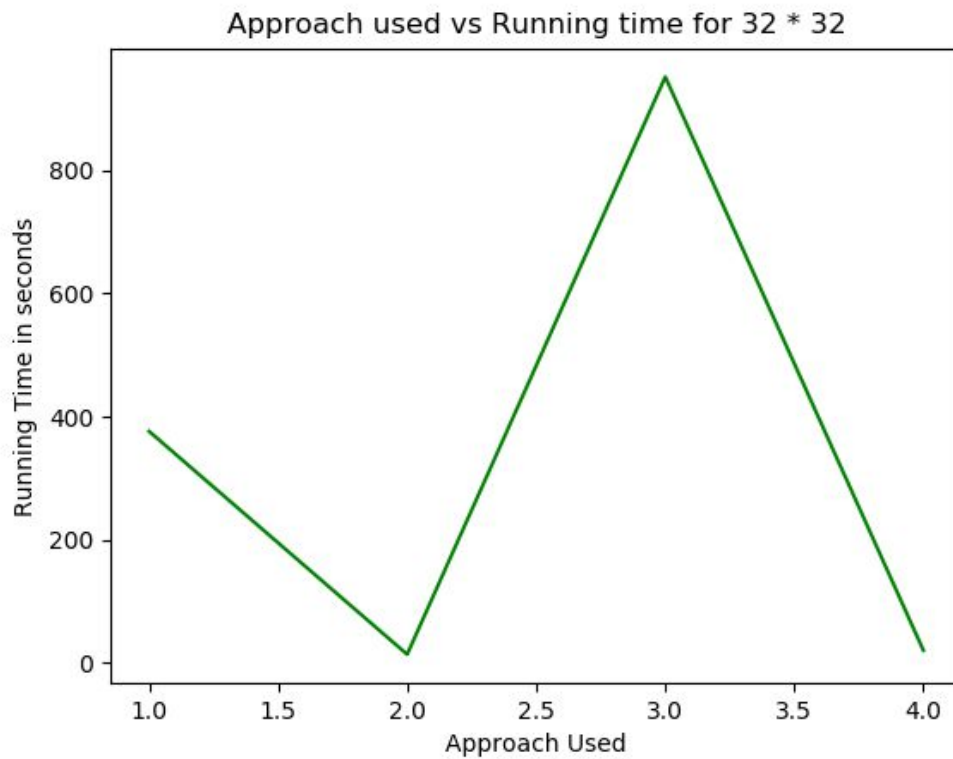
Question1.

Following table summarizes the findings for the first question.

	MapSize = 8 * 8	MapSize = 16* 16	MapSize = 32 * 32
Running Time: App1	0.734	19.275	376.068
Running Time:App2	0.293	2.674	14.12
Running Time:App3	11.384	62.284	951.161
Running Time:App4	1.14	4.12	20.642

Table: Running times for different approaches on different map sizes





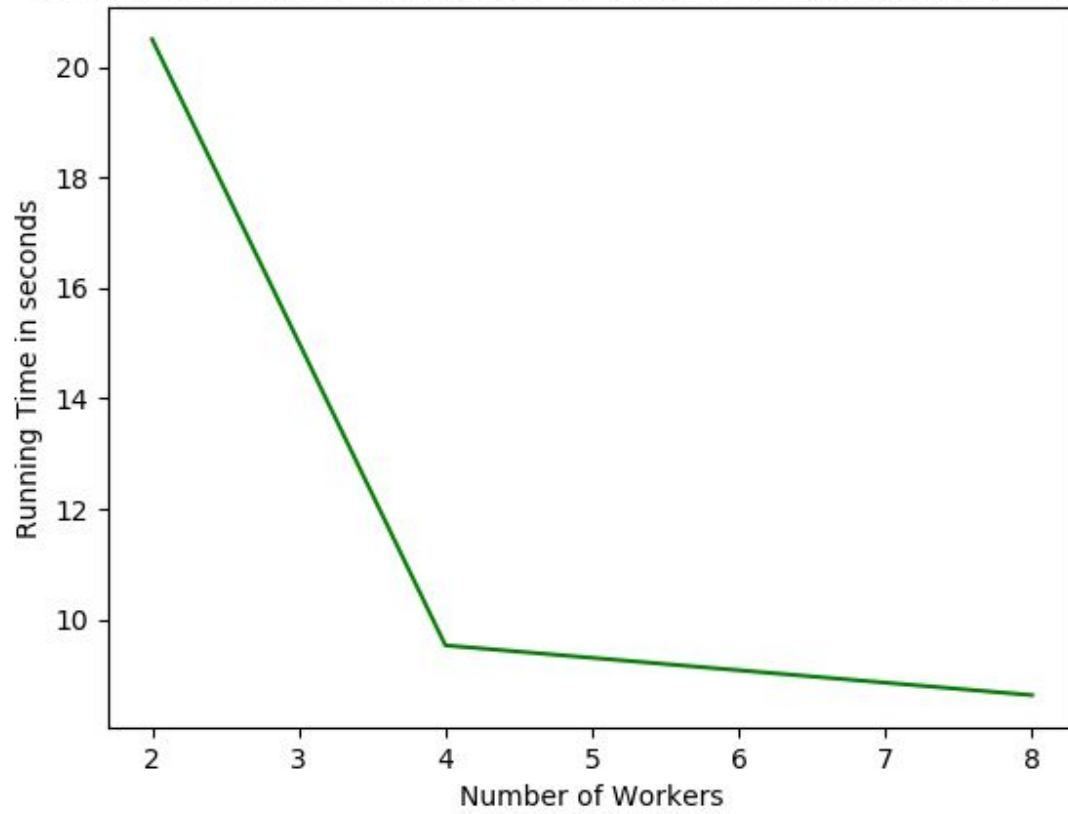
Question2.

	Workers = 2	Workers = 4	Workers = 8
Approach 3, 8 * 8	20.50	9.53	8.63
Approach 3, 16 * 16	143.3	63.58	53.73
	Workers = 2	Workers = 4	Workers = 8
Approach 4, 8 * 8	1.50	2.84	2.26
Approach 4, 16 * 16	3.33	4.23	4.86

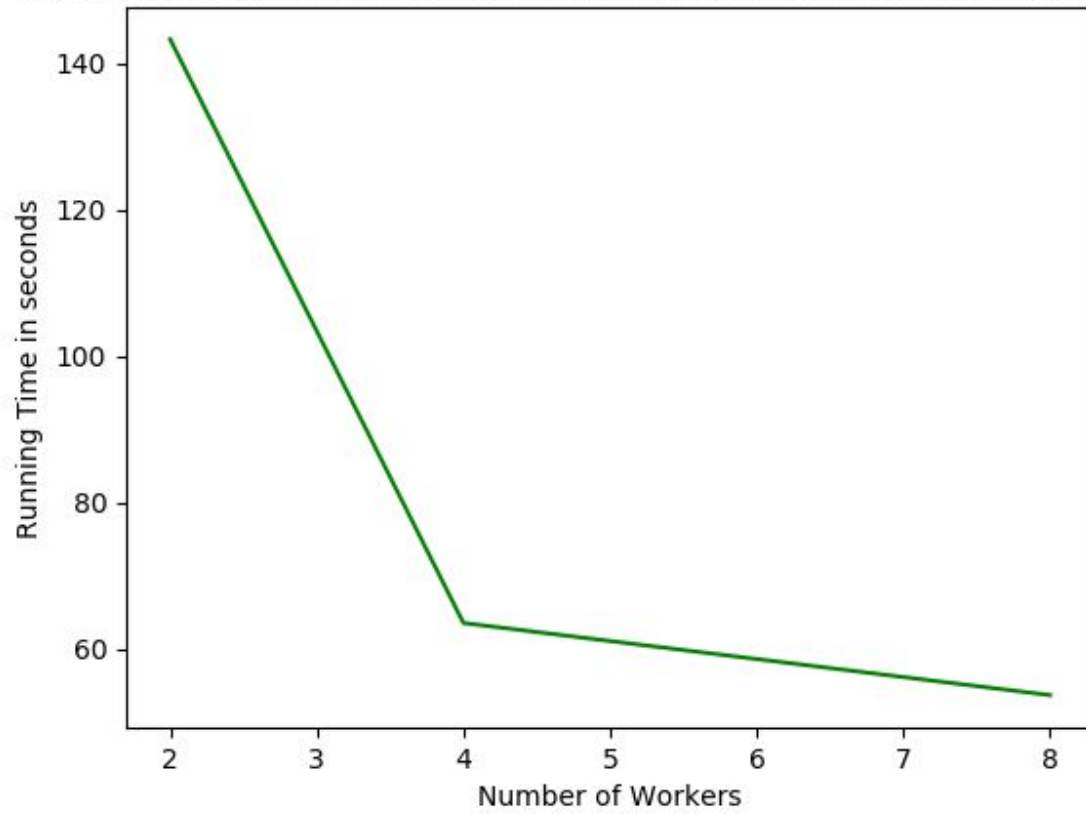
Note: The connection died when we tried this part for 32 * 32 and we skipped it because the question does not ask for 32 * 32.

Corresponding graphs follow.

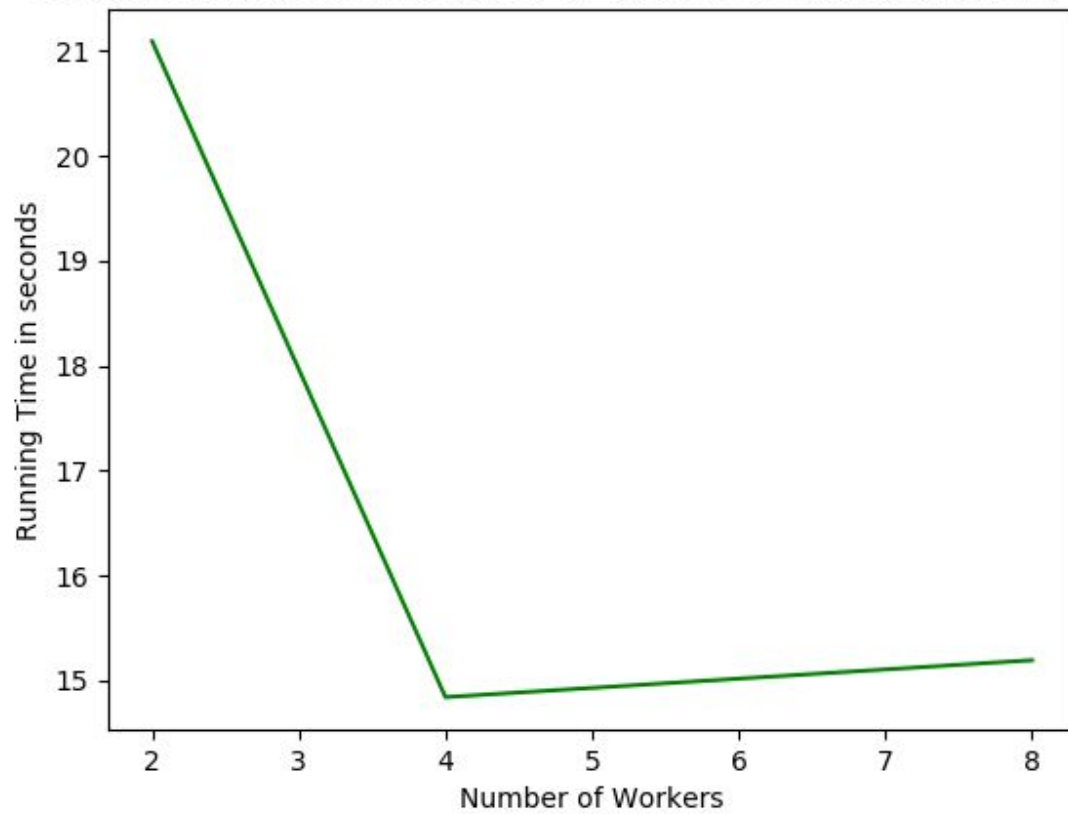
Distributed Approach 1: Number of Workers vs Running time for $8 * 8$



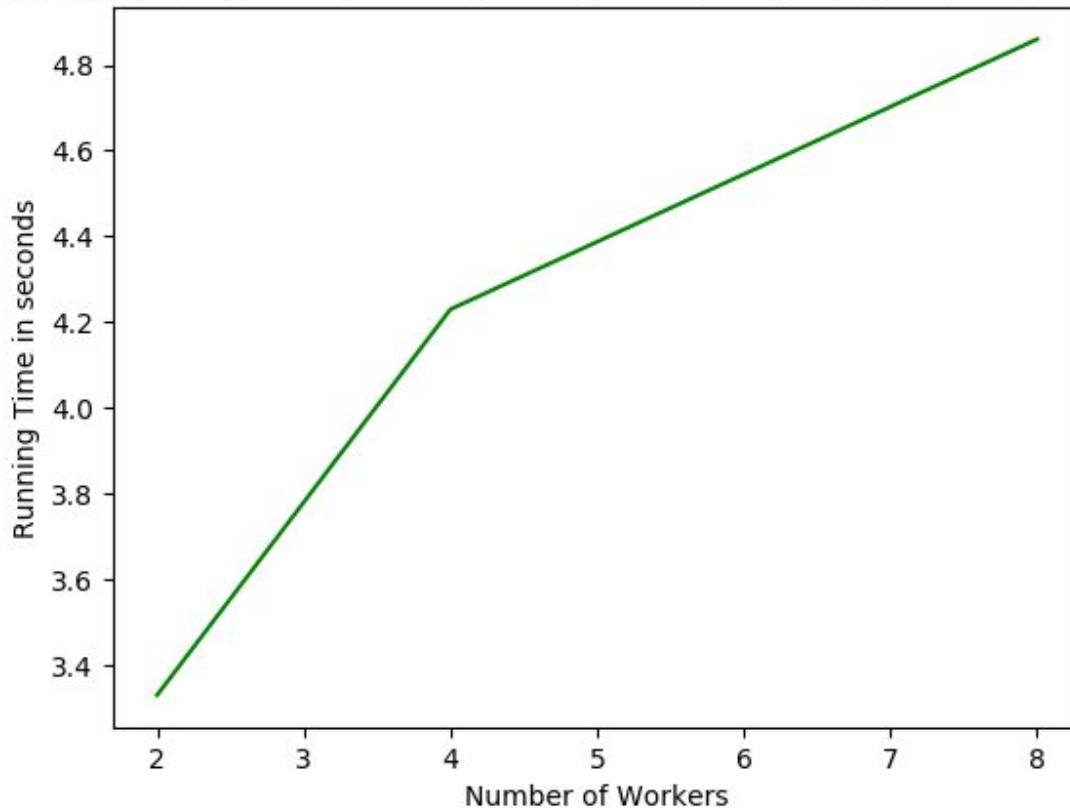
Distributed Approach 1: Number of Workers vs Running time for $16 * 16$



Distributed Approach 2: Number of Workers vs Running time for $8 * 8$



Distributed Approach 2: Number of Workers vs Running time for $16 * 16$



Question3:

The first distributed implementation creates n different processes for n states. However, the second distributed implementation creates just 4 different processes. Every time a process needs to update the value of a state in the first implementation, it needs to communicate to all other $(n-1)$ states because that is the only way to collect all the required value functions to calculate the value function for its own state. However, in the second case, since there are only 4 workers, a worker needs to communicate with only 3 other workers to receive all the value functions required. Hence, this significantly reduces the communication overhead and makes the whole pipeline faster.

Question4:

(Answering in terms of $8 * 8$ map according to the first table since we made multiple runs for the images)

The best distributed is the version 2 approach. 1.14 seconds

The best non-distributed is the version 2 approach. 0.293 seconds

The non-distributed version is better. It is because in $8 * 8$ map, the amount of computation required is not much. Hence, the communication overhead in the distributed approach actually overshadows the time required for actual computation.. However, if you see the trend of time consumption, as the size of the map increases, the gap between these decreases and eventually, the distributed implementation takes over.